
Programme de Formation

C++ : Maîtriser la POO pour concevoir des logiciels robustes

Description

Formation pratique pour maîtriser la syntaxe C++, les principes de la POO et développer des applications robustes et performantes avec les meilleures pratiques du langage.

Points Forts

Développez vos compétences en C++ à travers des ateliers pratiques couvrant syntaxe, gestion mémoire, POO et normes modernes. Apprenez à concevoir des applications performantes en appliquant les bonnes pratiques de développement professionnel.



Public visé

- Développeurs
- Ingénieurs logiciels
- Chefs de projets techniques proches du développement



Objectifs pédagogiques

- Comprendre la syntaxe et les concepts fondamentaux du C++
- Maîtriser les ajouts majeurs des normes C++
- Appliquer les principes de la Conception Orientée Objet
- Ecrire des programmes simples en appliquant les bonnes pratiques de développement
- Utiliser les structures de contrôle et les types de données en C++
- Manipuler les fichiers et la mémoire de manière basique

Organisation

Durée : 35 heures

Mode d'organisation : Mixte

Contenu pédagogique



Description

Jour 1 :

Ecrire des programmes simples en appliquant les bonnes pratiques de développement

- Historique C++
- Les différents compilateurs
- Environnements de développement
- Spécification et corps de main
- Affichage de valeurs et de chaînes avec cout
- Lecture des valeurs avec cin
- Formatage des sorties avec des manipulateurs de flots
- Exemples de travaux pratiques : le participant doit écrire, compiler et exécuter un premier programme C++ permettant à l'utilisateur de saisir son prénom et de dire « Bonjour » à l'utilisateur.



Utiliser les structures de contrôle et les types de données en C++

A l'issue de cette séquence, vous êtes capable d'écrire un programme utilisant des variables numériques et des opérateurs numériques.

- Déclaration et initialisation de variable
- Types de données entiers
- Types de données à virgule flottante
- Calcul arithmétique et affichage des résultats
- Mélange des types entiers et à virgule flottante dans les calculs et affectations
- Utilisation des références pour l'efficacité et des constantes pour la sécurité
- Exemples de travaux pratiques : le participant écrit un programme calculant la surface d'un triangle avec la formule de Héron.

Jour 2 :

Comprendre la syntaxe et les concepts fondamentaux du C++

- Passage des arguments aux fonctions et retour des valeurs depuis des fonctions
- Passage des arguments : par valeur ou par référence
- Visibilité, durée et valeur initiale des variables temporaires locales et des paramètres
- Exemples de travaux pratiques : le participant doit écrire une fonction calculant une vitesse et une deuxième échangeant deux valeurs.

Maîtriser la syntaxe du langage C++

- Prise de décision avec if/else
- Valeurs logiques bool ou int
- Chaînes d'instructions if/else
- Exécution de boucles avec while et for
- Opérateurs d'égalité, relationnel et logique
- Opérateurs incrémentiels et décréments
- Exemples de travaux pratiques : le participant écrit un programme implémentant l'algorithme d'Euclide.

Jour 3 :

Manipuler les fichiers et la mémoire de manière basique

- Déclaration et utilisation de tableaux et pointeurs
- Stockage de chaînes dans des tableaux de caractères
- Accès aux éléments des tableaux par l'intermédiaire des pointeurs
- Pointeurs ou références
- Classe string standard et ses méthodes
- Exemples de travaux pratiques : le participant doit écrire une fonction qui permet de vérifier si un mot est un palindrome.

Maîtriser les ajouts majeurs des normes C++

- Les conteneurs de la librairie standards
- Les constructeurs
- Les itérateurs
- Les méthodes
- Les algorithmes
- Exemples de travaux pratiques : le participant doit écrire un programme manipulant une liste de villes et un ensemble ordonné de température.

Jour 4 :

Appliquer les principes de la Conception Orientée Objet (1/2)

- Classe et objets
- Membres publics et privés
- Attributs et méthodes
- Constructeurs et destructeurs
- Auto-référence : le pointeur this
- Exemples de travaux pratiques : le participant doit écrire une classe représentant un point en 2D avec un nom et des coordonnées x, y.

Appliquer les principes de la Conception Orientée Objet (2/2)

- Séparation en interface et implémentation
- Signature des méthodes
- Surcharge des opérateurs
- Méthode amie
- Affectation par copie, constructeur par copie
- Méthodes en ligne et constexpr
- Exemples de travaux pratiques : le participant doit surcharger les opérateurs de comparaison de la classe Point2D.

Jour 5 :**Concevoir des applications C++ utilisant des classes (1/2)**

- Classes de base et dérivée
- Réutilisabilité par extensions incrémentielle
- Redéfinition des méthodes parentes
- Méthodes virtuelles
- Méthodes virtuelles pures
- Exemples de travaux pratiques : le participant doit écrire une classe décrivant un point pondéré dérivée de la classe Point2D.

Concevoir des applications C++ utilisant des classes (2/2)

- Manipulation des objets en mode direct ou par référence
- Passage des objets par valeur, par référence en mode constant ou pas
- Pile et tas
- Allocation dynamique et désallocation
- Pointeurs et tableaux d'objets
- Exemples de travaux pratiques : le participant doit écrire une fonction d'allocation d'un tableau de points 2D et une fonction d'affichage.

 Prérequis

- Connaître les principes de la programmation orientée objet
- Disposer d'une expérience pratique dans au moins un langage de programmation (C, Java, Python, etc.)
- Un questionnaire de positionnement sera réalisé au préalable pour adapter la formation à vos besoins

**Modalités pédagogiques**

- Exercices concrets
- Cas pratiques
- Quiz d'évaluation des connaissances



Moyens et supports pédagogiques

- Mise à disposition de postes de travail
- Vidéoprojecteur, tableau blanc, supports numériques
- Documentation pédagogique numérique



Modalités d'évaluation et de suivi

- Positionnement en amont de la formation :
 - Un quiz de consolidation des pré-requis sera administré en amont de la formation
- Suivi « pendant » :
 - Feuilles de présence
 - Exercices pratiques
 - Évaluation « fin de formation »
 - Évaluation des acquis en fin de formation
 - Formulaire d'évaluation de la formation
- Évaluation à froid :
 - Suivi post-formation : Questionnaire de satisfaction à j+30